# ***Curso HTML 5 / CSS 3 Java Script***

Curso Gustavo Guanabara

Sumário

[***Curso HTML 5 / CSS 3 Java Script*** 1](#_Toc195196995)

[**HTML 5***(HyperTest Markup Language)* 9](#_Toc195196996)

[***Estrutura Básica de documento em HTML 5*** 9](#_Toc195196997)

[***Caracteres Especiais em HTML 5*** 9](#_Toc195196998)

[**Caracter < (less than)** 9](#_Toc195196999)

[**Caracter > (greater than)** 9](#_Toc195197000)

[**Espaço em branco ( )** 9](#_Toc195197001)

[**Símbolo de Marca Registrada** 9](#_Toc195197002)

[**Símbolo de Marca Copyright** 9](#_Toc195197003)

[**Símbolo de Trade Mark** 9](#_Toc195197004)

[**Símbolo de Euro** 9](#_Toc195197005)

[**Símbolo de Libra** 9](#_Toc195197006)

[**Símbolo de Yen** 9](#_Toc195197007)

[**Símbolo de Cent** 10](#_Toc195197008)

[**Símbolo de Vazio Ø** 10](#_Toc195197009)

[**Símbolo de Soma** 10](#_Toc195197010)

[**Símbolo de Delta** 10](#_Toc195197011)

[**Símbolo de Setas ← ↑ → ↓** 10](#_Toc195197012)

[***TAGS em HTML 5*** 11](#_Toc195197013)

[**TAGS <!-- --> (comentário)** 11](#_Toc195197014)

[**TAGS <main>...</main>** 11](#_Toc195197015)

[**TAGS <h?> </h?> (Títulos)** 11](#_Toc195197016)

[**TAGS <p> </p> (parágrafo)** 11](#_Toc195197017)

[**TAGS <hr> (linha horizontal) - não tem TAG de fechamento** 11](#_Toc195197018)

[**TAGS <br> (quebra de linha) - não tem TAG de fechamento** 11](#_Toc195197019)

[**Emoji** 11](#_Toc195197020)

[**Inserindo Imagens (<img>)** 11](#_Toc195197021)

[**Ícone de favorito (<link rel= “shortcut icon” href= “favicon.ico” type = “image/x-icon”>)** 12](#_Toc195197022)

[**TAGS <div>...</div>** 12](#_Toc195197023)

[***Formatação de textos em HTML*** 12](#_Toc195197024)

[**Negrito <strong> e *Itálico <em>*** 12](#_Toc195197025)

[**Marca Texto <mark>...</mark>** 12](#_Toc195197026)

[**<big> morreu, mas o <small> ainda sobrevive** 12](#_Toc195197027)

[**Texto deletado** 12](#_Toc195197028)

[**Texto inserido** 12](#_Toc195197029)

[**Textos sobrescrito e subscrito** 13](#_Toc195197030)

[**Trechos de código** 13](#_Toc195197031)

[**“Citações”** 13](#_Toc195197032)

[**Abreviações** 14](#_Toc195197033)

[**ODITREVNI OTXET** 14](#_Toc195197034)

[***LISTAS em HTML 5*** 14](#_Toc195197035)

[**Listas Ordenadas <ol>...</ol>** 14](#_Toc195197036)

[**Listas Não Ordenadas <ul>...</ul>** 15](#_Toc195197037)

[**Listas de definição <dl>...</dl>** 15](#_Toc195197038)

[***Links e Âncoras em HTML 5*** 15](#_Toc195197039)

[**Links externos tag <a>: <a href="link">…</a>** 15](#_Toc195197040)

[**Links internos tag <a>: <a href="link">…</a>** 16](#_Toc195197041)

[**Links para downloads tag <a>: <a href="link">…</a>** 16](#_Toc195197042)

[https://www.iana.org/assignments/media-types/media-types.xhtml 16](#_Toc195197043)

[***Imagens Dinâmicas, Áudio e Vídeos*** 17](#_Toc195197044)

[**Imagens flexíveis <picture>...</picture>** 17](#_Toc195197045)

[**Áudios <audio>...</audio>** 18](#_Toc195197046)

[**Vídeos <video>...</videoo>** 19](#_Toc195197047)

[***GROUPINT TAGS E SEMANTIC TAGS*** 19](#_Toc195197048)

[**TAGS <div>...</div>** 20](#_Toc195197049)

[**TAGS <span>...</span>** 20](#_Toc195197050)

[**TAGS <header>...</header>** 20](#_Toc195197051)

[**TAGS <main>...</main>** 20](#_Toc195197052)

[**TAGS <footer>...</footer>** 20](#_Toc195197053)

[**TAGS <nav>...</nav>** 20](#_Toc195197054)

[**TAGS <section>...</section>** 20](#_Toc195197055)

[**TAGS <article>...</article>** 20](#_Toc195197056)

[**TAGS <aside>...</aside>** 20](#_Toc195197057)

[CSS (Cascading Style Sheets) 21](#_Toc195197058)

[***Style Tipos de inserir*** 21](#_Toc195197059)

[**Códigos inline Style** 21](#_Toc195197060)

[**Códigos Internal Style** 21](#_Toc195197061)

[**Códigos External Style** 22](#_Toc195197062)

[***SELETORES*** 23](#_Toc195197063)

[**fontes** 23](#_Toc195197064)

[**@font-face { }** 23](#_Toc195197065)

[**\* { }** 24](#_Toc195197066)

[**a { }** 24](#_Toc195197067)

[**Body { }** 24](#_Toc195197068)

[**h1 { }** 24](#_Toc195197069)

[**h2 { }** 24](#_Toc195197070)

[**h1, h2 { }** 24](#_Toc195197071)

[**p { }** 24](#_Toc195197072)

[***SELETORES PERSONALIZADOS*** 24](#_Toc195197073)

[**h1#principal { }** 25](#_Toc195197074)

[**.topico { }** 25](#_Toc195197075)

[***PSEUDO CLASSES*** 25](#_Toc195197076)

[**a: active { }** 25](#_Toc195197077)

[**a: hover { }** 25](#_Toc195197078)

[**a: visited { }** 25](#_Toc195197079)

[**div: hover { }** 25](#_Toc195197080)

[**div > p { }** 25](#_Toc195197081)

[**div: hover > p { }** 25](#_Toc195197082)

[***PSEUDO ELEMENTOS*** 27](#_Toc195197083)

[**a:: after { }** 27](#_Toc195197084)

[**a:: before { }** 27](#_Toc195197085)

[**.especial a:: after { }** 27](#_Toc195197086)

[**.especial a:: before { }** 27](#_Toc195197087)

[***ELEMENTOS DE CAIXA*** 28](#_Toc195197088)

[***COMANDOS DE CAIXA EM CSS 3*** 29](#_Toc195197089)

[**Shorthand Font** 29](#_Toc195197090)

[**background-color: lightsteelblue;** 29](#_Toc195197091)

[**border-width: 10px;** 29](#_Toc195197092)

[**border-style: solid;** 29](#_Toc195197093)

[**border-color: darkgray;** 29](#_Toc195197094)

[**Shorthand border: 10px darkgray solid;** 29](#_Toc195197095)

[**height: 300px;** 29](#_Toc195197096)

[**padding-top: 10px;** 29](#_Toc195197097)

[**padding-right: 10px;** 29](#_Toc195197098)

[**padding-bottom: 10px;** 29](#_Toc195197099)

[**padding-left: 10px;** 29](#_Toc195197100)

[**Shorthand padding: 10px 10px 10px 10px;** 29](#_Toc195197101)

[**margin: auto;** 29](#_Toc195197102)

[**margin-top: 20px;** 29](#_Toc195197103)

[**margin-right: 20px;** 30](#_Toc195197104)

[**margin-bottom: 40px;** 30](#_Toc195197105)

[**margin-left: 20px;** 30](#_Toc195197106)

[**Shorthand margin: 20px 20px 40px 20px;** 30](#_Toc195197107)

[**Shorthand margin: 10px auto 40px auto;** 30](#_Toc195197108)

[**width: 300px;** 30](#_Toc195197109)

[**outline-width: 5px;** 30](#_Toc195197110)

[**outline-style: dashed;** 30](#_Toc195197111)

[**outline-color: salmon;** 30](#_Toc195197112)

[**Shorthand outiline: 5px dashed salmon;** 30](#_Toc195197113)

[***COMANDOS EM CSS 3*** 31](#_Toc195197114)

[**background-color: lightsteelblue;** 31](#_Toc195197115)

[**background-image: linear-gradient (to right, white, blue, yellow);** 31](#_Toc195197116)

[**background-image: url(…);** 31](#_Toc195197117)

[**background-position: left top;** 31](#_Toc195197118)

[**background-repeat: repeat;** 31](#_Toc195197119)

[**background-size: auto;** 31](#_Toc195197120)

[**background-attachment: fixed;** 31](#_Toc195197121)

[**Shorthand background: black url(…) center center repeat cover fixed;** 31](#_Toc195197122)

[**border: 1px solid black;** 31](#_Toc195197123)

[**border-image-source: url ( ‘…’ );** 31](#_Toc195197124)

[**border-image-slice: 20;** 31](#_Toc195197125)

[**border-image-repeat: repeat;** 32](#_Toc195197126)

[**Shorthand border-image: url ( ‘…’ ) 20 repeat;** 32](#_Toc195197127)

[**border-radius: 10px; 20px; 30px; 40px;** 32](#_Toc195197128)

[**color: darkblue;** 32](#_Toc195197129)

[**colums: 2;** 32](#_Toc195197130)

[**contente: “ ”**  ; 32](#_Toc195197131)

[**display: inline-block;** 32](#_Toc195197132)

[**box-shadow: inset 10px 10px 10xp 10px black;** 32](#_Toc195197133)

[**font-family: Arial, Helvetica, sans-serif;** 32](#_Toc195197134)

[**font-size: 20px;** 33](#_Toc195197135)

[**font-style: normal;** 33](#_Toc195197136)

[**font-variant: small-caps;** 33](#_Toc195197137)

[**font-weight: normal;** 33](#_Toc195197138)

[**line-height: 1.5em;** 33](#_Toc195197139)

[**List-style-position: inside;** 33](#_Toc195197140)

[**List-style-type: “simbolo”;** 33](#_Toc195197141)

[**Shorthand Font** 33](#_Toc195197142)

[**height: 100%;** 33](#_Toc195197143)

[**padding: 10px;** 33](#_Toc195197144)

[**padding-top: 10px;** 33](#_Toc195197145)

[**padding-right: 10px;** 33](#_Toc195197146)

[**padding-bottom: 10px;** 34](#_Toc195197147)

[**padding-left: 10px;** 34](#_Toc195197148)

[**Shorthand padding** 34](#_Toc195197149)

[**margin: auto;** 34](#_Toc195197150)

[**min-width: 300px;** 34](#_Toc195197151)

[**min-height: 150px;** 34](#_Toc195197152)

[**max-width: 1000px;** 34](#_Toc195197153)

[**max-height: 1000px;** 34](#_Toc195197154)

[**text-align: justify;** 34](#_Toc195197155)

[**text-decoration: underline;** 34](#_Toc195197156)

[**text-indent: 20px;** 34](#_Toc195197157)

[**text-shadow: 1px 1px 2xp black;** 34](#_Toc195197158)

[**text-transform: uppercase;** 34](#_Toc195197159)

[**transition-duration: 1s;** 35](#_Toc195197160)

[**width: 600px;** 35](#_Toc195197161)

[**transform: trnslate(-50%, -50%);** 35](#_Toc195197162)

[***VARIÁVEIS EM CSS 3*** 35](#_Toc195197163)

[**:root { }** 35](#_Toc195197164)

[***VÍDEO CENTRALIZADO*** 35](#_Toc195197165)

[***CENTRALIZAÇÃO VERTICAL DE CAIXAS*** 38](#_Toc195197166)

[**JAVASCRIPT** 39](#_Toc195197167)

[***Variáveis*** 39](#_Toc195197168)

[***Manipulação de Dados*** 39](#_Toc195197169)

[**Number(n) - Number.parseInt(n) – Number.parseFloat(n)** 39](#_Toc195197170)

[**String(n) – n.toString( )** 39](#_Toc195197171)

[***COMANDOS EM JAVASCRIPT*** 40](#_Toc195197172)

[***//*** 40](#_Toc195197173)

[***/\* ... \*/*** 40](#_Toc195197174)

[***Window.alert ( ‘...’ );*** 40](#_Toc195197175)

[***Window.confirm ( ‘...’ );*** 40](#_Toc195197176)

[***Window.prompt ( ‘...’ );*** 40](#_Toc195197177)

[**typeof** 40](#_Toc195197178)

[**Concatenação** 40](#_Toc195197179)

[**Template strings** 40](#_Toc195197180)

[**.lengh** 40](#_Toc195197181)

[**.toUpperCase( )** 40](#_Toc195197182)

[**.toLowerCase( )** 40](#_Toc195197183)

[**document.write(...)** 40](#_Toc195197184)

[**document.write(...)** 41](#_Toc195197185)

[***Formatação de Números*** 41](#_Toc195197186)

[**.toFixed(2)** 41](#_Toc195197187)

[**.toFixed(2).replace( ‘.’ , ‘,’)** 41](#_Toc195197188)

[**n1.toLocaleString(‘pt-BR’,{style: ‘currency’, currency:’BRL’})** 41](#_Toc195197189)

[**n1.toLocaleString(‘pt-BR’,{style: ‘currency’, currency:’USD’})** 41](#_Toc195197190)

[**n1.toLocaleString(‘pt-BR’,{style: ‘currency’, currency:’EUR’})** 41](#_Toc195197191)

[***Operadores*** 41](#_Toc195197192)

[**aritméticos ( + , - , \* , / , % , \*\* )** 41](#_Toc195197193)

[**Relacionais ( > , < , >= , <= , == , != )** 42](#_Toc195197194)

[**Lógicos ( ! , && , || )** 42](#_Toc195197195)

[**Ternário ( ? : )** 42](#_Toc195197196)

[***DOM*** 43](#_Toc195197197)

[**window.document.write(p1.innerText)** 43](#_Toc195197198)

[**window.document.write(p1.innerhtml)** 43](#_Toc195197199)

[***Eventos DOM*** 44](#_Toc195197200)

[***Funções JavaScript*** 44](#_Toc195197201)

[***Condições*** 44](#_Toc195197202)

[***Bibliografias*** 46](#_Toc195197203)

[**Material curso Gustavo Guanabara** 46](#_Toc195197204)

[https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/01%20-%20Hist%C3%B3ria%20da%20Internet.pdf 46](#_Toc195197205)

[https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/02%20-%20Como%20funciona%20a%20Internet.pdf 46](#_Toc195197206)

[https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/03%20-%20Como%20funciona%20HTML%20e%20CSS.pdf 46](#_Toc195197207)

[https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/04%20-%20Primeiros%20passos%20HTML.pdf 46](#_Toc195197208)

[https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/05%20-%20Caracteres%2C%20par%C3%A1grafos%20e%20quebras%20de%20linha.pdf 46](#_Toc195197209)

[https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/06%20-%20Imagens%20e%20Favicon.pdf 46](#_Toc195197210)

[https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/07%20-%20Hierarquia%20de%20t%C3%ADtulos.pdf 46](#_Toc195197211)

[https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/08%20-%20Formata%C3%A7%C3%A3o%20de%20Textos.pdf 46](#_Toc195197212)

[https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/09%20-%20Listas%20HTML.pdf 46](#_Toc195197213)

[https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/10%20-%20Liga%C3%A7%C3%B5es%20em%20toda%20parte.pdf 46](#_Toc195197214)

[https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/11%20-%20Imagens%20din%C3%A2micas%2C%20%C3%A1udios%20e%20v%C3%ADdeos.pdf 46](#_Toc195197215)

[https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/12%20-%20Trabalhando%20com%20estilos.pdf 46](#_Toc195197216)

[https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/13%20-%20Cores.pdf 46](#_Toc195197217)

[https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/14%20-%20Fontes.pdf 46](#_Toc195197218)

[https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/15%20-Seletores%20personalizados.pdf 47](#_Toc195197219)

[https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/16%20-%20Modelo%20de%20Caixas.pdf 47](#_Toc195197220)

[https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/17%20-%20Primeiro%20mini-projeto.pdf 47](#_Toc195197221)

[https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/18%20-%20Aprendendo%20Git%20e%20GitHub.pdf 47](#_Toc195197222)

[https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/19%20-%20Imagens%20de%20Fundo.pdf 47](#_Toc195197223)

[https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/20%20-%20Mini-projeto%20Cordel.pdf 47](#_Toc195197224)

[**Referêcias on-line** 48](#_Toc195197225)

[**Livros** 48](#_Toc195197226)

[**Programas** 49](#_Toc195197227)

[https://www.gimp.org/ 49](#_Toc195197228)

[https://handbrake.fr/ 49](#_Toc195197229)

[**Imagens Direitos Autorais** 50](#_Toc195197230)

[https://unsplash.com/pt-br/ 50](#_Toc195197231)

[https://www.pexels.com/pt-br/ 50](#_Toc195197232)

[**Ícone de favoritos** 50](#_Toc195197233)

[https://www.iconarchive.com/ 50](#_Toc195197234)

[https://www.favicon.cc/ 50](#_Toc195197235)

[https://favicon.io/ 50](#_Toc195197236)

[**Efeitos e músicas** 50](#_Toc195197237)

[**Vídeos** 50](#_Toc195197238)

[https://www.pexels.com/pt-br/ 50](#_Toc195197239)

[**Listas de media types** 51](#_Toc195197240)

[https://www.iana.org/assignments/media-types/media-types.xhtml 51](#_Toc195197241)

[**Cores** 51](#_Toc195197242)

[https://color.adobe.com/pt/ 51](#_Toc195197243)

[https://paletton.com/#uid=1000u0kllllaFw0g0qFqFg0w0aF 51](#_Toc195197244)

[https://coolors.co/ 51](#_Toc195197245)

[**Fontes** 51](#_Toc195197246)

[https://fonts.google.com/ 51](#_Toc195197247)

[https://www.dafont.com/pt/ 51](#_Toc195197248)

[https://www.whatfontis.com/ 51](#_Toc195197249)

[https://www.fontsquirrel.com/ 51](#_Toc195197250)

[https://www.myfonts.com/pages/tags/my-fonts 51](#_Toc195197251)

# **HTML 5***(HyperTest Markup Language)*

## ***Estrutura Básica de documento em HTML 5***

Arquivo inicial 🡪 index.html

Para criar automaticamente o código básico no Visual Studio Code, digitar na linha 1 ( ! )

*<!DOCTYPE html>*

*<html lang=”pt-br”>*

*<head>*

*<meta charset=”UTF-8”>*

*<meta name = “viewport” content = “width=device-width, initial-scale=1.0”>*

*<title> Document </title>*

*</head>*

*<body>*

*<h1> Olá, Mundo! </h1>*

*</body>*

*</html>*

## ***Caracteres Especiais em HTML 5***

### **Caracter < (less than)**

&lt – insere o símbolo <

### **Caracter > (greater than)**

&gt – insere o símbolo >

### **Espaço em branco ( )**

&nbsp – insere um espaço em branco



### **Símbolo de Marca Registrada**

&reg – insere o símbolo de marca registrada



### **Símbolo de Marca Copyright**

&copy – insere o símbolo de marca copyright



### **Símbolo de Trade Mark**

&trade – insere o símbolo de trade mark

### **Símbolo de Euro**

&euro – insere o símbolo de Euro

### **Símbolo de Libra**

&pound – insere o símbolo de libra



### **Símbolo de Yen**

&yen – insere o símbolo de Yen



### **Símbolo de Cent**

&cent – insere o símbolo de Cent

### **Símbolo de Vazio Ø**

&empty – insere o símbolo de Vazio



### **Símbolo de Soma**

&sum – insere o símbolo de Soma



### **Símbolo de Delta**

&Delta – insere o símbolo de Delta

### **Símbolo de Setas ← ↑ → ↓**

&larr – insere seta para esquerda

&uarr – insere seta para cima

&rarr – insere seta para direita

&darr – insere seta para baixo

## ***TAGS em HTML 5***

### **TAGS <!-- --> (comentário)**

<!-- Inserir comentários -->

### **TAGS <main>...</main>**

Conteúdo principal

### **TAGS <h?> </h?> (Títulos)**

Comandos de abertura de TAG (<\_\_>) e Fechamento de TAG (</\_\_>). Exemplo <h1>Conteúdo </h1>.

Títulos

<h1> Conteúdo </h1> <h2> Conteúdo </h2> <h3> Conteúdo </h3>

<h4> Conteúdo </h4> <h5> Conteúdo </h5> <h6> Conteúdo </h6>

### **TAGS <p> </p> (parágrafo)**

<p> parágrafo </p>

### **TAGS <hr> (linha horizontal) - não tem TAG de fechamento**

<hr> cria uma linha horizontal

### **TAGS <br> (quebra de linha) - não tem TAG de fechamento**

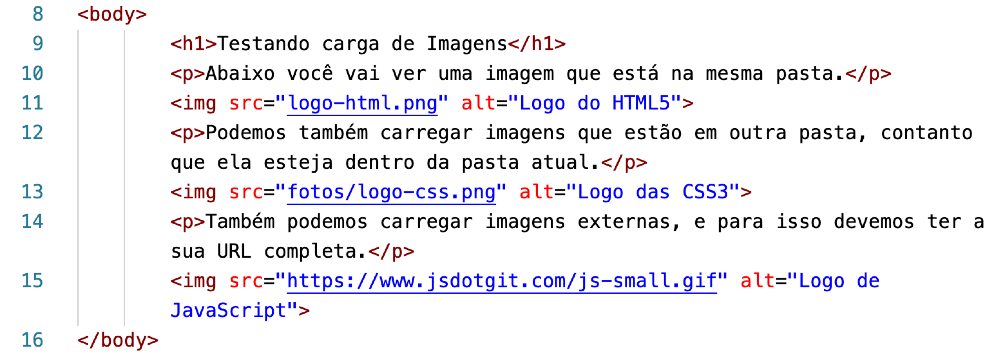
<br> quebra a linha

### **Emoji**

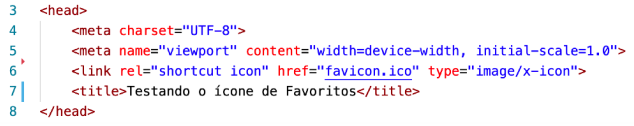
Para inserir o Emoji, escolhe no site da <https://emojipedia.org/> e insere o código do emoji com o comando **&#x**1F596; atenção o 1F596 é o código pelo pego site do emoji escolhido.

### **Inserindo Imagens (<img>)**

Formato mais usado JPEG e PNG <img src=" " alt=" " /> (src = origem da imagem) (alt = texto explicativo da imagem)



### **Ícone de favorito (<link rel= “shortcut icon” href= “favicon.ico” type = “image/x-icon”>)**



### **TAGS <div>...</div>**

<div>... </div> É um espaço qualquer

## ***Formatação de textos em HTML***

### **Negrito <strong> e *Itálico <em>***

A tag <strong> significa que o termo delimitado possui força dentro da frase. Logo, ele aparecerá em negrito.

Já a tag <em> significa que queremos dar ênfase (do Inglês emphasis) ao termo. Logo, ele aparecerá em itálico.

Note que, ao usar <strong> e <em> no lugar de <b> e <i>, damos mais significado aos nossos termos e conteúdo. Como eles vão ser representados visualmente (forma), vai depender das nossas folhas de estilo CSS.

### **Marca Texto <mark>...</mark>**

Para fazer essas marcas em HTML5, usamos a tag <mark>…</mark> para delimitar o texto que queremos demarcar, como se estivéssemos usando uma caneta marcador.

### **<big> morreu, mas o <small> ainda sobrevive**

Tem certas coisas que simplesmente não são fáceis de entender. A tag <big> (que deixava o texto maior) está depreciada pela HTML5, mas a tag <small> (que deixa o texto menor> segue firme e forte na vida da linguagem.

### **Texto deletado**

Antigamente existia a tag <strike>, que hoje também está depreciada pela última versão da HTML. No lugar dela entrou a tag <del>, que significa que o texto está ali, pode até ser lido, mas deve ser desconsiderado pelo leitor. Exatamente como eu acabei de fazer.

### **Texto inserido**

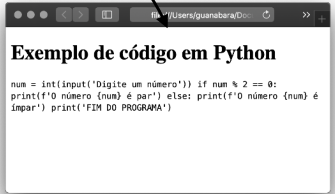
O texto inserido é o exato oposto do texto deletado que vimos anteriormente. Nesse caso, se colocarmos um texto qualquer dentro de <ins> e </ins>, estamos dizendo que o texto está ali, deve ser lido e você deve prestar atenção nele. É o que fazíamos anteriormente ao utilizar a tag <u> para sublinhar um texto.

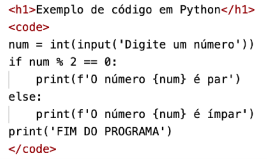
### **Textos sobrescrito e subscrito**

Quantas vezes você tentou escrever um x2 ou um H2O em um documento e ficou na dúvida de como fazer pra esses números ficarem pequeninos ali em cima ou embaixo. Pois a HTML tem as tags <sub> e <sup> justamente para essa finalidade.

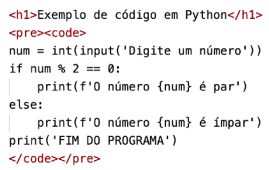
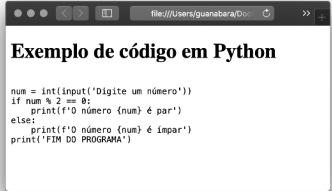
### **Trechos de código**

Nós somos (ou seremos) programadores, e por isso compartilhamos muito código-fonte em diversas linguagens. Para isso, existe a tag <code> da HTML onde você pode delimitar seu código. A principal vantagem no uso dessa tag é a o valor semântico que ela representa, indicando ao navegador que se trata de um código de computador. Porém, existe também um efeito visual, pois as letras ficam no modo mono-espaçadas (monospace), o que facilita bastante a leitura do código. Analise, por exemplo, o trecho de código a seguir:





Para resolver esse problema, vamos usar uma outra tag HTML chamada <pre>, que mantém o texto pré-formatado, exatamente da mesma maneira na qual ele foi digitado, incluindo quebras de linhas, espaços e tabulações.

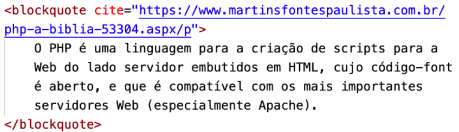


Notou agora a diferença? A junção das tags <pre> e <code> em conjunto nos trouxe um resultado visual bem mais interessante.

### **“Citações”**

Uma citação é um trecho de texto, escrito ou dito por outra pessoa, que vai ilustrar perfeitamente algo que você quer explicar. Normalmente, uma citação aparece entre aspas ou com as margens deslocadas em relação ao texto.

Para criar uma citação em HTML, podemos usar a tag <q> (do Inglês quote, que significa citar). O texto que estiver entre <q> e </q> já vai receber automaticamente as aspas, mas não terá nenhum deslocamento. Essa técnica é mais usada quando queremos uma citação no meio de um parágrafo.

Também podemos criar citações mais longas (em bloco) e que tenham um parágrafo só para si. Nesse caso, colocaremos tudo dentro de <blockquote> e </blockquote> e o texto ganha um recuo automaticamente. Podemos também colocar um link para o texto original, usando o parâmetro cite dentro da tag.

### **Abreviações**

Essa é uma novidade da HTML5 e que ajuda muito em áreas como a de Tecnologia, que usa muitas siglas e abreviações. Sempre que você quiser escrever uma sigla, mas deixar claro ao usuário (e aos mecanismos de busca) o significado dela, use a tag <abbr>.



Note no código acima que usamos o title para indicar o significado da sigla. Ao lado, mostro o funcionamento desse código no navegador.

Quando passamos o mouse sobre a sigla abreviada, um pequeno texto aparece com o seu significado.



### **ODITREVNI OTXET**

Não entendeu nada desse título? Pois volte ali em cima e leia da última letra até a primeira. Achou inútil? Eu também. Pois é exatamente essa a função da tag <bdo>. Para começo de conversa, BDO significa bi-directional override. Ao usar essa tag, coloque também o parâmetro dir para indicar uma das duas direções possíveis:

rtl = da direita para a esquerda (right-to-left)

ltr = da esquerda para a direita (left-to-right)



## ***LISTAS em HTML 5***

### **Listas Ordenadas <ol>...</ol>**

Listas Ordenadas com índice preenchidos automaticamente.

<ol>

<li>Acordar</li>

<li>Ligar para o João</li>

<li>Tomar café</li>

<li>Escovar os dentes</li>

</ol>

O fechamento </li> no html5 é opcional.

Adicionar n tags iguais: <tag>\*n

Parâmetro Type (opções 1, A, a, I, i) 🡪 <ol type="1" start"4"> start numeração inicial da lista

### **Listas Não Ordenadas <ul>...</ul>**

Listas Não Ordenadas <ul>: com marcadores.

<ul>

<li>Pão</li>

<li>Leite</li>

<li>Tomate</li>

<li>Alface</li>

</ul>

Parâmetro Type (opções square, disc, circle) 🡪 <ul type="square">

### **Listas de definição <dl>...</dl>**

A tag <dl> em HTML é usada para criar uma lista de definições, que é uma lista de pares de termos e descrições.

É comumente usada para implementar um glossário ou exibir metadados (uma lista de pares de chave e valor.

<dl>

<dt>HTML</dt>

<dd>Linguagem de Marcação de Hipertexto</dd>

<dt>CSS</dt>

<dd>Folhas de Estilo em Cascata</dd>

</dl>

Neste exemplo, HTML e CSS são os termos (definidos pela tag <dt>) e Linguagem de Marcação de Hipertexto e Folhas de Estilo em Cascata são as descrições correspondentes (definidas pela tag <dd>).

## ***Links e Âncoras em HTML 5***

### **Links externos tag <a>: <a href="link">…</a>**

Abrir link em nova aba: <a href="link" target="\_blank" rel="external"> </a>

onde:

href= “...” entre as aspas é a referência do link hipertexto URL

target = “\_blank” abre em uma nova aba do navegador

rel = indica qual a natureza do destino e aceita vários valores como:

author 🡪 indica que é um link para o site do autor do artigo atual

external 🡪 indica que é um link para outro site que não faz parte do site atual

nofollow 🡪 indica que é um link para um site não endossado, como um link pago



Quando não tiver o link para digitar, colocar # que significa link vazio.

### **Links internos tag <a>: <a href="link">…</a>**

Abrir link em nova aba: <a href="link" target="\_blank" rel="external"> </a>

onde:

target =“\_blank” abre em uma nova aba do navegador

target =“\_self” abre na própria aba

rel = indica qual a natureza do destino e aceita vários valores como:

next 🡪 indica que o link é para a próxima parte do documento atual

prev 🡪 indica que o link é para a parte anterior do documento atual

author 🡪 indica que é um link para o site do autor do artigo atual



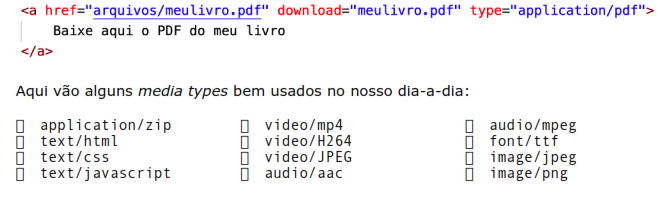
Caso o arquivo esteja dentro de outra pasta, indica o caminho ()

Para voltar para a pagina anterior, tem que indicar tambem o caminho “../nome do arquivo que quer abrir”

Quando não tiver o link para digitar, colocar # que significa link vazio.

### **Links para downloads tag <a>: <a href="link">…</a>**

Outra coisa que aparece bastante em sites são os links para efetuar download de algum material em PDF, ou de um arquivo ZIP qualquer. A partir da versão HTML5, as âncoras receberam atributos especiais para isso. Basta fazer o link diretamente para o arquivo que se deseja efetuar o download e adicionar o atributo download com o valor configurado para o nome do arquivo a ser baixado e o atributo type para indicar ao navegador que tipo de arquivo está sendo baixado.



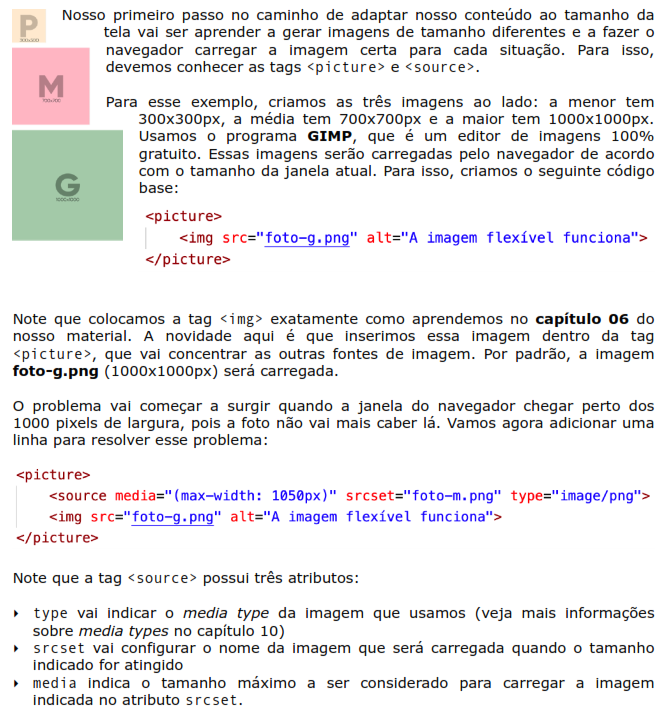
LISTA DE MEDIA TYPES: Se você quer saber o que escrever dentro do atributo type de uma âncora de hypertext, consulte a lista oficial da IANA.org disponível no link abaixo.

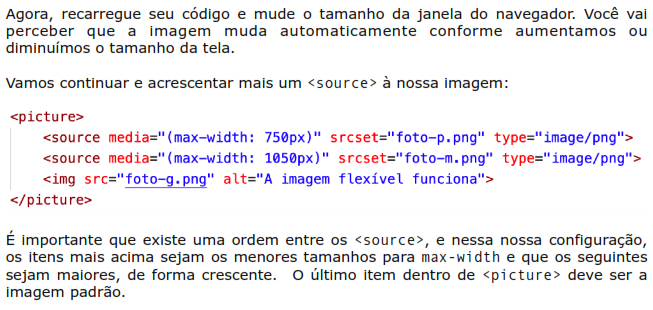
Quando não tiver o link para digitar, colocar # que significa link vazio.

<https://www.iana.org/assignments/media-types/media-types.xhtml>

## ***Imagens Dinâmicas, Áudio e Vídeos***

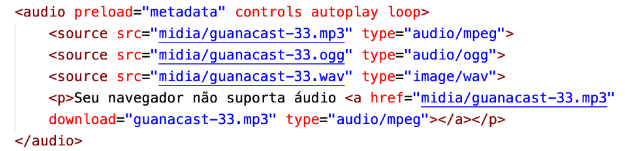
## **Imagens flexíveis <picture>...</picture>**





## **Áudios <audio>...</audio>**

Vamos analisar os principais atributos da tag <audio> antes de mais nada.



O atributo preload indica se o áudio será pré-carregado ou não e aceita três valores:

๏ metadata vai carregar apenas as informações sobre o arquivo (tamanho, tempo, informações de direitos, etc).

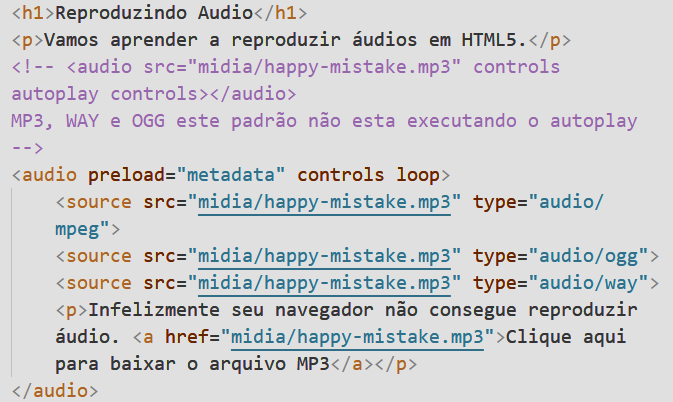
๏ none não vai carregar absolutamente nada até que o usuário clique no botão play ou um script inicie a reprodução.

๏ auto (padrão) vai carregar o arquivo de áudio inteiro assim que a página for carregada, mesmo que o usuário nunca aperte o play.

‣ O atributo controls vai apresentar o player na tela. Caso não seja colocado na tag <audio>, o controle será transparente e o usuário não poderá interagir com ele.

‣ O atributo autoplay, quando inserido, vai iniciar a reprodução do áudio assim que a página for carregada.   
‣ O atributo loop vai fazer com que o áudio seja repetido eternamente assim que terminar a sua reprodução.

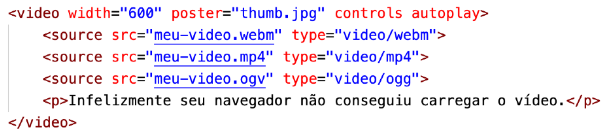
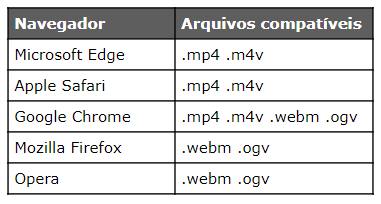
Dentro da tag <audio>, adicionamos vários <source> com formatos diferentes do mesmo áudio. Coloque na parte de cima o seu formato favorito. Os demais só serão carregados caso o de cima falhe. Caso todos falhem, criamos um parágrafo que permite o download do arquivo MP3 para ouvir no player padrão do dispositivo.



## **Vídeos <video>...</videoo>**

Para inserir um vídeo em nosso site, podemos utilizar a nova tag <video> da HTML5, caso o arquivo esteja hospedado no nosso próprio servidor.

Formatos de arquivos



Antes de mais nada, vamos criar a tag <video> e configurar alguns atributos importantes:

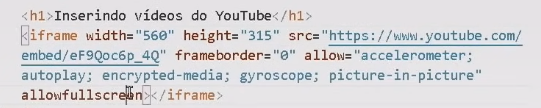
‣width vai indicar a largura que o vídeo vai ter na tela. Nesse exemplo, 600px.

‣poster configura uma imagem que vai aparecer como uma capa, enquanto o visitante não aperta o play para reproduzir o vídeo.

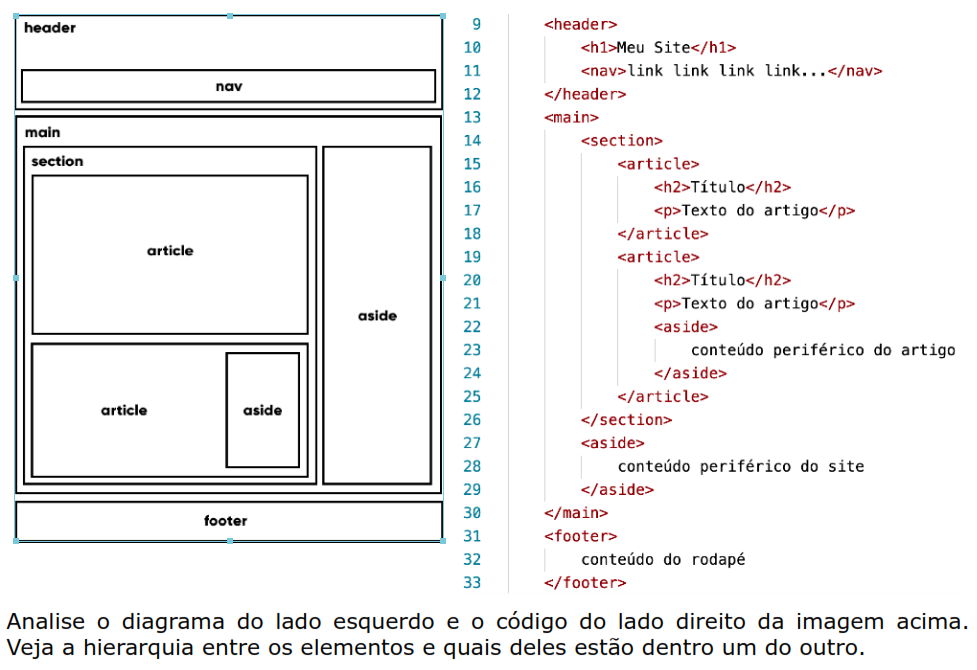
‣controls vai configurar se os controles do vídeo vão aparecer na parte inferior da mídia. Por padrão, os controles não aparecerão, mas basta colocar a palavra controls na tag <video>.

‣autoplay diz para o navegador se o vídeo vai começar a tocar automaticamente, assim que a página for carregada.

Para não consumir dados do cliente, utilizar os vídeos do YouTube (free) ou Vímeo (pago), selecionar o vídeo e clicar em compartilhar e escolher a opção incorporar e inserir o código no programa.



## ***GROUPINT TAGS E SEMANTIC TAGS***



### **TAGS <div>...</div>**

Caixa tipo box level

### **TAGS <span>...</span>**

Caixa tipo inlinelevel

### **TAGS <header>...</header>**

Cabeçalho

### **TAGS <main>...</main>**

Conteúdo principal

### **TAGS <footer>...</footer>**

Rodapé

### **TAGS <nav>...</nav>**

Links

### **TAGS <section>...</section>**

Seção

### **TAGS <article>...</article>**

Artigo

### **TAGS <aside>...</aside>**

links

## ***Tabelas***

### **TAGS <table>...</table>**

Delimita minha tabela

Hierarquia de tabelas (simples)

table = tabela

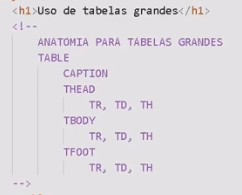
table row = linha da tabela (tr)

table header = cabeçalho de tabela

table data = dado de tabela (td)

### **TAGS <tr>...</tr>**

Table row 🡪 criando a primeira linha da tabela. Fechamento da tag tr e td são opcionais



<table>

<tr>

<td>A1</td>

<td>B1</td>

<td>C1</td>

</tr>

</table>

# **CSS (Cascading Style Sheets)**

Utiliza-se seletores com declarações e sempre tem o ponto e vírgula (;) no final

## ***Style Tipos de inserir***

Em termos de configuração o que prevalece primeiro é: Inline, interno e por último o externo.

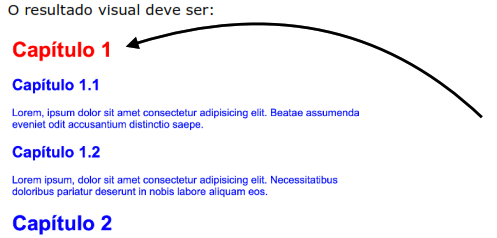
## **Códigos inline Style**

Pouco usual, mas possível. A alteração é feita em linha para cada tag HTML.



Muito cuidado na hora de digitar esse código. Tudo deve ser seguido exatamente como fizemos acima, inclusive com letras maiúsculas e minúsculas. Não esqueça de adicionar os ponto e vírgulas para separar as declarações. Note que o formato da letra mudou (era Times e ficou em Arial) e a cor da fonte também foi alterado para azul.





Note que apenas o Capítulo 1 ficou vermelho, o Capítulo 2 - que também é um <h1> - não teve alteração alguma. Isso acontece pois estamos fazendo configurações pontuais usando CSS.

## **Códigos Internal Style**

Usual, A alteração é feita dentro da tag head no HTML e todos os estilos são colocados dentro da tag style.

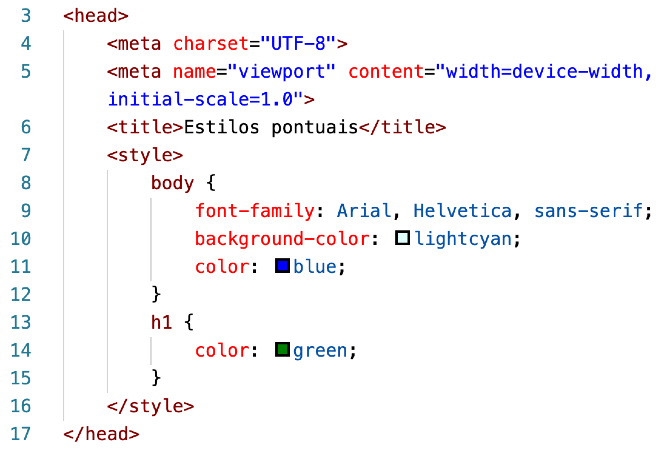
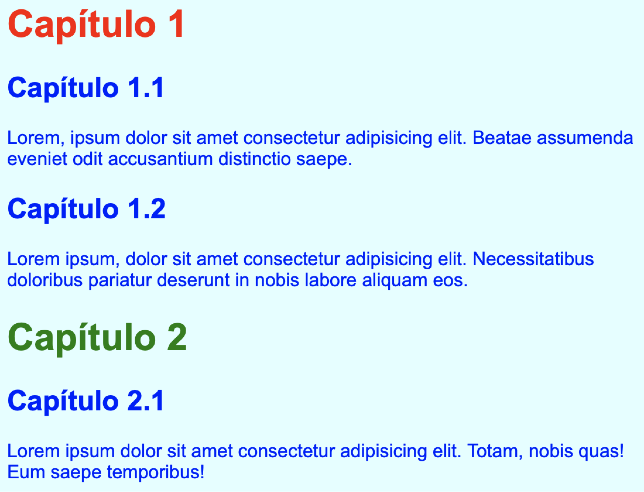
Ponto negativo, se você tiver muitas páginas e tiver que alterar os estilos de todas, você terá que alterar em todas as páginas.

Para aplicar estilos de forma mais dinâmica e prática, podemos adicionar uma tag <style> dentro da área <head> do nosso documento HTML local. Volte lá no seu VSCode, e adicione o código dentro de <head>.

**ATENÇÃO!** A tag <style> deve estar dentro da área <head> do seu documento HTML5. Se você a colocar em qualquer outro local, como dentro da tag <body>, o resultado até pode funcionar, mas seu código estará fora dos padrões estabelecidos pela W3C. Siga sempre as regras!

Você sabe explicar por que o Capítulo 1 ficou vermelho e não verde, como solicitamos?

Isso acontece porque as configurações pontuais (HTML style) vão prevalecer sobre as configurações gerais (CSS style). Volte ao seu código e remova todas as configurações de estilo que fizemos nas tags <body> e <h1> no início do capítulo.



## **Códigos External Style**

Manter as folhas de estilo fora do código HTML, além de uma maior organização faz com que tudo seja reaproveitado de maneira mais eficiente nas outras páginas do nosso site. Para isso, utilizamos a tag <link> especialmente configurada para trabalhar com arquivos externos de estilo. Essa tag deve ser colocada dentro da área <head> do seu documento HTML.



Arquivo style.css



Na linha 1, colocamos uma regra em CSS, que vai indicar a compatibilidade de codificação com o padrão UTF-8, assim como fizemos com o arquivo HTML5. Essa linha não é obrigatória e normalmente nem vai aparecer na maioria dos seus arquivos de configurações de estilo, mas caso você comece a ter problemas de compatibilidade com alguns caracteres, saiba que ela existe.

Na linha 2, adicionamos um comentário para facilitar a documentação do arquivo.

Os comentários - assim como vimos em HTML - só servirão de explicação para que o desenvolvedor entenda o funcionamento de uma determinada linha ou trecho de código. O navegador não vai considerar nada que está entre os símbolos /\* e \*/ em CSS.

Nas demais linhas, fizemos as configurações dos seletores, da mesma maneira que criamos com as outras duas técnicas apresentadas no capítulo.

## ***SELETORES***

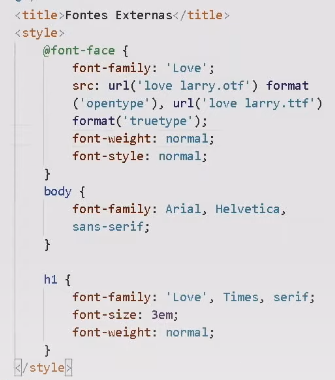
## **fontes**

Importando fonte do site da GooGle Fonts

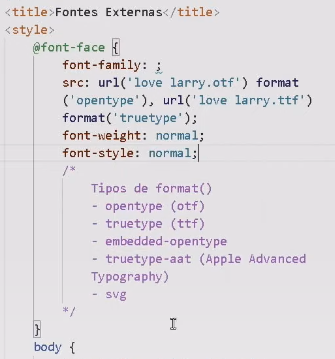
ATENÇÃO : não pode estar dentro de nenhum seletor, deve estar logo abaixo de Style



## **@font-face { }**



Fonte foi baixada do site dafonte.com





## **\* { }**

Seletor global da CSS

## **a { }**

Tudo que for colocado dentro das chaves vai ser aplicado nas âncoras do HTML

## **Body { }**

Tudo que for colocado dentro das chaves vai ser aplicado no corpo do HTML

## **h1 { }**

Tudo que for colocado dentro das chaves vai ser aplicado no título h1 do HTML

## **h2 { }**

Tudo que for colocado dentro das chaves vai ser aplicado no título h1 do HTML

## **h1, h2 { }**

Tudo que for colocado dentro das chaves vai ser aplicado no título h1, h2 do HTML

## **p { }**

Tudo que for colocado dentro das chaves vai ser aplicado no parágrafo do HTML

## ***SELETORES PERSONALIZADOS***

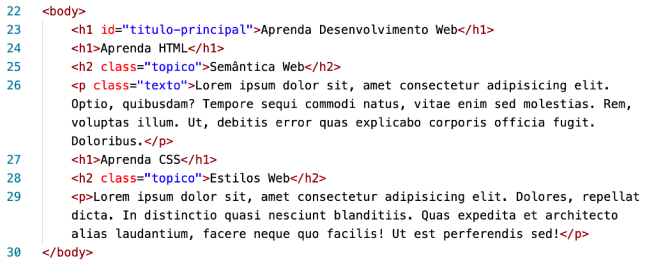
***# = id***

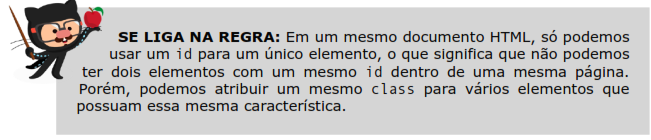
***. = class***

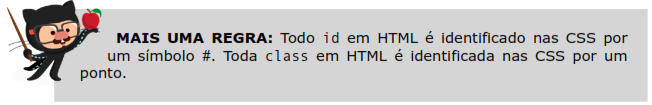
***: = pseudo-class***

***:: = pseudo-element***

***> = clildren***







## **h1#principal { }**

Tudo que for colocado dentro das chaves vai ser aplicado apenas na tag h1 id= “principal”

## **.topico { }**

Tudo que for colocado dentro das chaves vai ser aplicado apenas nas tags com class=“tópico”

Podemos colocar mais de um class na mesma tag separados por espaço

## ***PSEUDO CLASSES***

Relacionado ao estado de um determinado elemento, tipo se ele está marcado, ativo, vazio, desativado, habilitado, quando passar o mouse por cima etc.

## **a: active { }**

É ativado assim que eu clicar com o mouse e segurar com o mouse <a> div.

Tudo que for colocado dentro das chaves vai ser aplicado no corpo da tag ao clicar e segurar.

## **a: hover { }**

É ativado assim que eu passar o mouse <a>.

Tudo que for colocado dentro das chaves vai ser aplicado no corpo da tag

## **a: visited { }**

É ativado assim que eu passar o mouse nesta de âncora <a> div.

Tudo que for colocado dentro das chaves vai ser aplicado no corpo da tag

## **div: hover { }**

É ativado assim que eu passar o mouse nesta div.

Tudo que for colocado dentro das chaves vai ser aplicado no corpo da tag

## **div > p { }**

Parágrafo (filho) que está dentro de uma div.

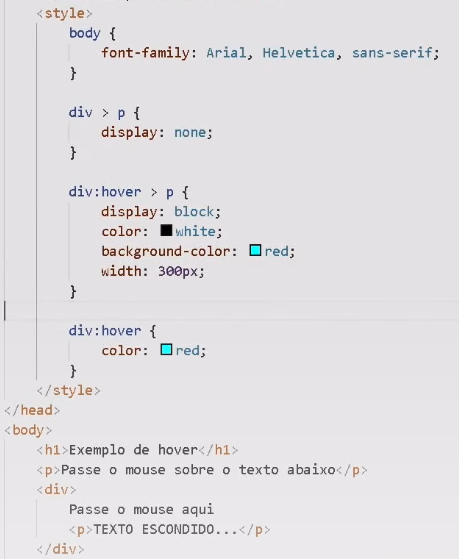
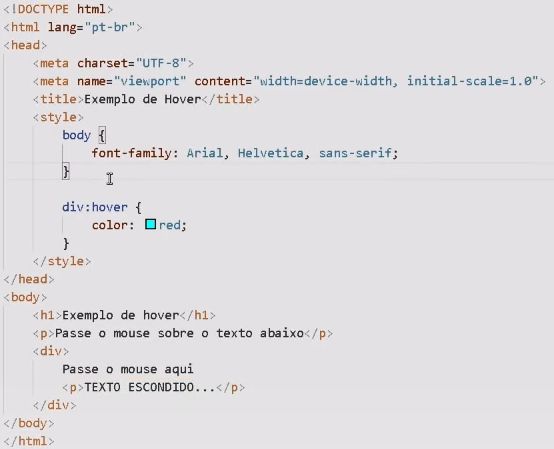
div tem um filho que é o parágrafo (p).

 faz desaparecer o texto do <p> dentro da div.

## **div: hover > p { }**

Altera o paragrafo (p) que está dentro de uma div.

div tem um filho que é o parágrafo (p).





**ESCLARECENDO:**

Nesse caso, você precisa lembrar dos conceitos de COMBINADORES DE SELETORES, que nada mais é do que criar uma relação útil entre os seletores.

Você usa o " > " quando quer indicar que deseja estilizar um FILHO DIRETO de um elemento. Segue um exemplo:

<header>

<h1></h1>

<p></p>

</header>

No caso do exemplo acima, o "p" é um filho direto da div especial <header>, e portanto é um elemento de Primeiro Nível. Da mesma forma, veja que o "h1" identicamente é um filho direto de header e portanto é um elemento de Primeiro Nível também. É como se <p> e <h1> fossem IRMÃOS.

Mas quando você NÃO USA o " > ", você quer indicar os elementos que não são necessariamente filhos, mas são DESCENDENTES. Veja outro exemplo:

<main>

<article>

<h1></h1>

</article>

</main>

*SELETOR NÃO SIMPLIFICADO PARA O EXEMPLO ACIMA:* " main > article > h1 {...} ". H1 é filho direto de article, que é filho direto de main.

Tem como simplificar? Sim. Nesse caso, perceba que h1 é descendente de main (como se fosse um NETO ), então você pode DEIXAR DE USAR o " > " sem problema nenhum, dessa forma estará indicando elementos que não são necessariamente filhos, e sim *descendentes.* Exemplo:

*MESMO SELETOR ANTERIOR, SÓ QUE SIMPLIFICADO:* " main h1 {...} "

Obs.: Se tivéssemos criado o seletor simplificado do exemplo acima, mas com o sinal " > " ( ex: main > h1 {...} ), não funcionaria porque esse h1 NÃO É FILHO DIRETO do main, e sim neto. Sendo que o sinal de maior ( ">" ) serve apenas para indicar filhos diretos.

## ***PSEUDO ELEMENTOS***

Pode agir nas classes nos elementos nos Ids e ele mexe no conteúdo periférico do elemento.

Utiliza-se :: para pseudo elementos (adiciona algo)

## **a:: after { }**

Depois que todos os links vou avisar que é um link

Ex: a::after{

content: ‘ [link]’;

text-decoration: none;

font-weight: normal;

color: darkgray;

}

## **a:: before { }**

Insere algo antes

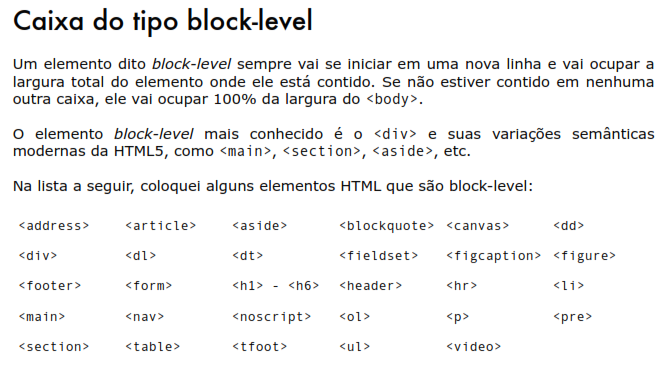
## **.especial a:: after { }**

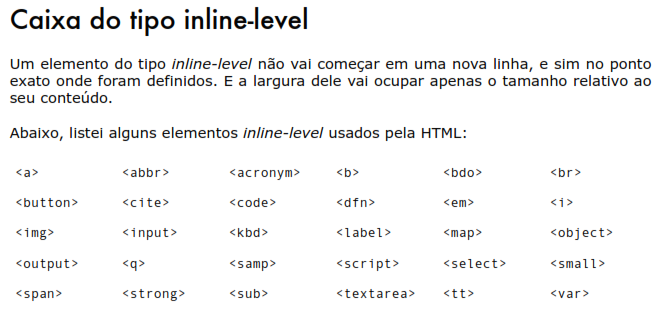
Insere algo depois com a classe especial

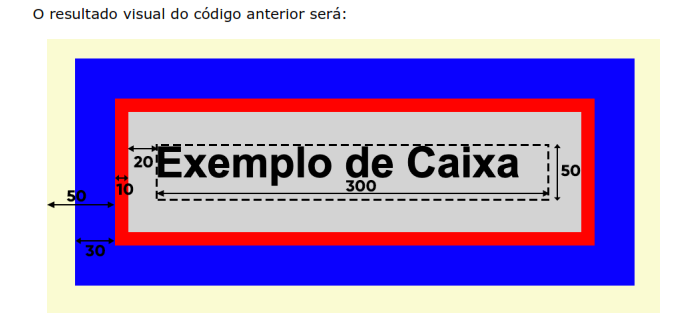
## **.especial a:: before { }**

Insere algo antes com a classe especial

## ***ELEMENTOS DE CAIXA***







## ***COMANDOS DE CAIXA EM CSS 3***

### **Shorthand Font**

Ordem (font-style 🡪 font-weight 🡪 font-size 🡪 font-family )

font: italic bolder 3em ‘Work Sans’, sans-serif;

### **background-color: lightsteelblue;**

Aplica a cor selecionada no fundo da caixa.

### **border-width: 10px;**

Largura da borda.

### **border-style: solid;**

Estilo da borda.

**solid**

**dashed**

**dotted**

**double**

**groove** (Estilo de borda em 3D)

### **border-color: darkgray;**

Cor da borda.

### **border-collapse: separate;**

**separate** 🡪 Linha da tabela separada.

**collapse** 🡪 Linha da tabela juntas.

### **Shorthand border: 10px darkgray solid;**

Ordem (border 🡪 width 🡪 style 🡪 color )

### **height: 300px;**

Altura da caixa.

### **padding-top: 10px;**

borda

### **padding-right: 10px;**

borda

### **padding-bottom: 10px;**

borda

### **padding-left: 10px;**

borda

### **Shorthand padding: 10px 10px 10px 10px;**

Ordem (padding 🡪 top 🡪 right 🡪 bottom 🡪 left )

Se todas os lados forem iguais, indicar somente 10px que já serve para todos os lados

Se colocar dois valores padding: 10px 20px; a de cima e a de baixo são iguais a 10px e a da esquerda e da direita vão ter 20px.

### **margin: auto;**

Centraliza a margem (caixa)

### **margin-top: 20px;**

Margem topo

### **margin-right: 20px;**

Margem direita

### **margin-bottom: 40px;**

Margem baixo

### **margin-left: 20px;**

Margem esquerda

### **Shorthand margin: 20px 20px 40px 20px;**

Ordem (padding 🡪 top 🡪 right 🡪 bottom 🡪 left )

Se todas os lados forem iguais, indicar somente 20px que já serve para todos os lados

Se colocar dois valores margin: 10px 20px; a de cima e a de baixo são iguais a 10px e a da esquerda e da direita vão ter 20px.

### **Shorthand margin: 10px auto 40px auto;**

Centraliza a margem da esquerda e da direita

### **width: 300px;**

Largura do conteúdo considerado na tela.

### **outline-width: 5px;**

Fica entre a borda e a margem mas dentro da margem

### **outline-style: dashed;**

Coloca a linha em estilo tracejado

### **outline-color: salmon;**

Coloca a cor da linha

### **Shorthand outiline: 5px dashed salmon;**

Ordem (outline 🡪 width 🡪 style 🡪 color )

## ***COMANDOS EM CSS 3***

### **background-color: lightsteelblue;**

Aplica a cor selecionada no fundo indicado pelo seletor.

### **background-image: linear-gradient (to right, white, blue, yellow);**

**Radial-gradient:**

Para fazer cores em degradê.

Pode usar to right, to left . 90deg, to top, to bottom, circle

Pode usar quantas cores quiser para fazer o degradê

Pode tambem usar uma cor e a segunda usar transparente ()transparent para fazer a cor iniciar e terminar transparente.

### **background-image: url(…);**

Coloca uma imagem de fundo com a url(‘...’), indicando o caminho da imagem nos ‘....’ .

A imagem só tem sentido visual de fundo.

Atenção a imagem é copiada uma do lado da outra.

Tamanho da imagem, cuidado com tamanho do arquivo e direitos autorais.

A imagem vai ser carregada no tamanho padra a não ser que eu coloque o background-size: 100px 100px; que dimensiona esta imagem pelo valor do px.

A imagem por padrão tambem é colocada pelo canto superior esquerdo.

### **background-position: left top;**

left top 🡪 É o valor padrão.

left center 🡪 A posição é centro pela esquerda.

right bottom 🡪 A posição é direita embaixo.

center center 🡪 A posição é o centro da foto.

### **background-repeat: repeat;**

repeat 🡪 É o valor padrão.

no-repeat 🡪 Fica com somente uma imagem

repeat-x 🡪 Repete a imagem na direão X

repeat-y 🡪 Repete a imagem na direção Y

### **background-size: auto;**

auto 🡪 É o valor padrão.

contain 🡪 Mostra a imagem 100% exibida e conforme eu diminuo a tela , vai aparecer bordas. Atenção não esquecer de colocar no style o height:100vh; para usar 100% da view port.

cover 🡪 Cobre a tela inteira, mas a imagem é adaptada para o tamanho da view port ou seja a imagem vai aparecer na tela inteira mas pode não mostrar o tamanho todo da imagem.

### **background-attachment: fixed;**

fixed 🡪 Deixa o fundo fixo na rolagem de tela.

scroll 🡪 O fundo rola junto com o conteúdo.

### **Shorthand background: black url(…) center center repeat cover fixed;**

Ordem (color 🡪 image 🡪 position 🡪 repeat 🡪 size 🡪 attachment)

Atenção [size] 🡪 apos colocar a shorthand o valor de size, a imagem sumiu ou seja ainda não esta aplicável o size na shorthand.

### **border: 1px solid black;**

Borda da tag.

### **border-image-source: url ( ‘…’ );**

**url (‘...’)** 🡪 nome do arquivo de borda

### **border-image-slice: 20;**

número da repetição

### **border-image-repeat: repeat;**



**repeat** 🡪 repete a borda **stretch** 🡪 puxa a linha

### **Shorthand border-image: url ( ‘…’ ) 20 repeat;**

Ordem (source 🡪 slice 🡪 repeat )

### **border-radius: 10px; 20px; 30px; 40px;**

Arredonda as pontas.

Pode tambem usar porcentagem ex border-radius: 50%;

### **color: darkblue;**

Altera a cor da fonte do texto.

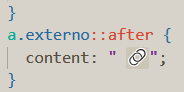
### **colums: 2;**

Altera em duas colunas.

6.0

### **contente: “ ”** ;

insere o símbolo ver ex abaixo:

 Insere a imagem depois dos links externos declarado com a classe externo.

### **display: inline-block;**

**inline-block** 🡪 Colocar tudo na mesma linha.

**none** 🡪 Não mostra o texto, esconde ele.

**block** 🡪 Mostra o texto.

### **box-shadow: inset 10px 10px 10xp 10px black;**

Coloca uma sombra na caixa.

1º parâmetro – Quanto que a sombra vai andar para o lado (horizontal)

2º parâmetro - Quanto que a sombra vai andar para baixo (vertical)

3º parâmetro - Quanto que a sombra vai embaçar

4º parâmetro - Quanto que a sombra vai espalhar

5º parâmetro - Quanto cor e transparência a sombra vão ter

**Outset** - default sombra interna, não precisa colocar

**inset** - sombra externa

### **font-family: Arial, Helvetica, sans-serif;**

Altera a fonte indicado pelo seletor.

Peso da fonte usar a fonte ‘Work Sans’ exemplo abaixo: ver **font-weight**:



### **font-size: 20px;**

Altera o tamanho da fonte indicado pelo seletor.

Medidas absolutas (cm, mm, in, **px**, pt, pc)

Medidas relativas (**em**, ex, rem, vw, vh, %)

Tamanho normal da fonte 16px

Geralmente 16px = 1em

### **font-style: normal;**

**italic**

**normal**

### **font-variant: small-caps;**

**small-caps** 🡪 transforma o texto em letras maiúscula, mas deixando a primeira letra maior.

**normal**

### **font-weight: normal;**

**lighter** (fino)

**normal** (normal)

**bold** (negrito)

**bolder** (espesso)

**100 a 900** (numérica começa em 100 e termina em 900 de 100 lighter e 900 bolder)

Não funciona em todas as fontes por não ter tanta variação de peso

### **line-height: 1.5em;**

Espaçamento entre linhas do texto digitado, o valor 1em é o texto sem espaçamento entre linhas**.**

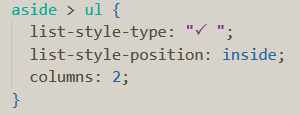
### **List-style-position: inside;**

Posição da lista dentro da caixa.

Normal é outside pelo lado de fora da caixa;

### **List-style-type: “simbolo”;**

Neste exemplo toda a lista dentro de aside o estilo vai ser alterado para o símbolo de ticket.



### **Shorthand Font**

Ordem (font-style 🡪 font-weight 🡪 font-size 🡪 font-family )

font: italic bolder 3em ‘Work Sans’, sans-serif;

### **height: 100%;**

Altura do conteúdo considerado na tela.

### **padding: 10px;**

borda de 10px em todos os lados

### **padding-top: 10px;**

borda cima

### **padding-right: 10px;**

borda direita

### **padding-bottom: 10px;**

borda embaixo

### **padding-left: 10px;**

borda esquerda

### **Shorthand padding**

Ordem (padding 🡪 top 🡪 right 🡪 bottom 🡪 left )

borda

### **margin: auto;**

Centraliza no meio da tela o conteúdo mesmo alterando a largura da tela.

Serve apenas para alinhamento horizontal.

### **min-width: 300px;**

Largura mínima da tela.

### **min-height: 150px;**

Altura mínima do box na tela.

### **max-width: 1000px;**

Largura máxima da tela.

### **max-height: 1000px;**

Altura máxima do box na tela.

### **text-align: justify;**

**justify;** 🡪 Alinha o texto indicado pelo seletor.

**center;** 🡪 Centraliza o texto indicado pelo seletor.

**left;** 🡪 Alinha o texto a esquerda indicado pelo seletor.

**right;** 🡪 Alinha o texto a direita indicado pelo seletor.

**right;** 🡪 Alinha o texto a direita indicado pelo seletor.

### **text-decoration: underline;**

**underline;** 🡪 Texto sublinhado.

**none;** 🡪 Retira o sublinhado do texto.

### **text-indent: 20px;**

Início do texto deslocado da palavra no parágrafo.

### **text-shadow: 1px 1px 2xp black;**

Coloca uma sombra no texto.

1º parâmetro – Quanto que a sombra vai andar para o lado

2º parâmetro - Quanto que a sombra vai andar para baixo

3º parâmetro - Quanto que a sombra vai espalhar

4º parâmetro - Quanto cor e transparência a sombra vai ter

### **text-transform: uppercase;**

**uppercase;** 🡪 Transforma o texto em letra maiúscula.

### **transition-duration: 1s;**

Duração da transição para acontecer o desejado.

### **vertical-align: top;**

**top;** 🡪 Os textos das células ficam para cima das células.

**bottom;** 🡪 Os textos das células ficam para baixo das células.

**middle;** 🡪 Os textos das células ficam no meio das células (padrão).

### **width: 600px;**

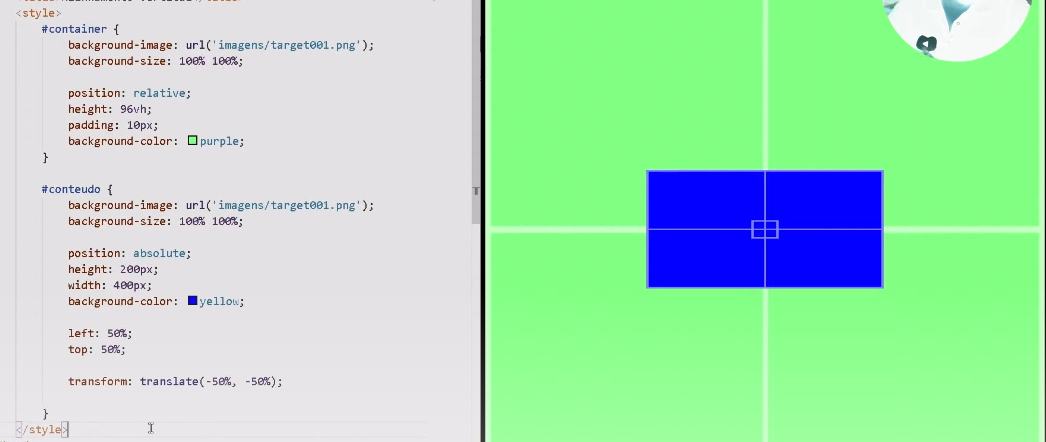
Largura do conteúdo considerado na tela.

### **transform: trnslate(-50%, -50%);**

Altera a posição de uma caixa dentro da outra para o centro.

A caixa container no posicionamento relativo e o conteúdo no posicionamento absoluto.

Deixa o left:50% e top:50% no conteúdo e então faz este comando de transform.



## ***VARIÁVEIS EM CSS 3***

### **:root { }**

Pseudo classe root (raiz), tudo que for colocada pode ser usado no documento inteiro.

Posso criar variáveis nesta área principal onde posso criar as variáveis globais.

Todas as variáveis em CSS começam com dois traços - - Ex;

-- cor0: #c5ebd6; 🡪 após os traços indica-se o nome

## ***VÍDEO CENTRALIZADO***

**HTML**

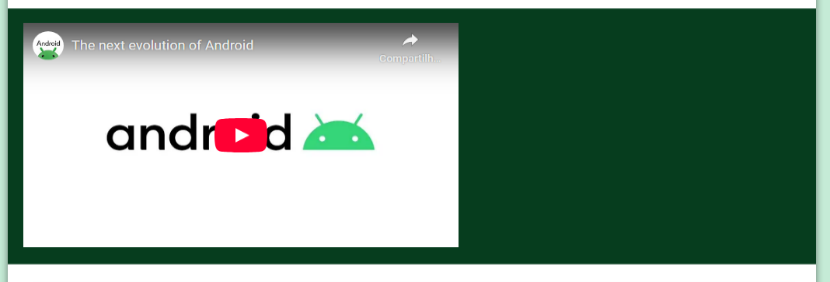
Passo 1 🡪 envelopar o iframe dentro de um div.

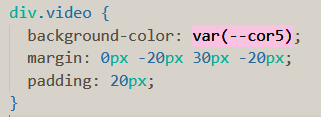
Passo 2 🡪 colocar esta div com uma classe (class=video). Class porque posso ter vários vídeos.

Passo 3 🡪 colocar posicionamento relativo

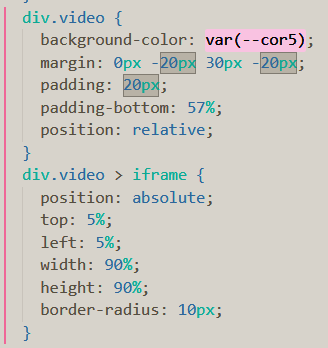
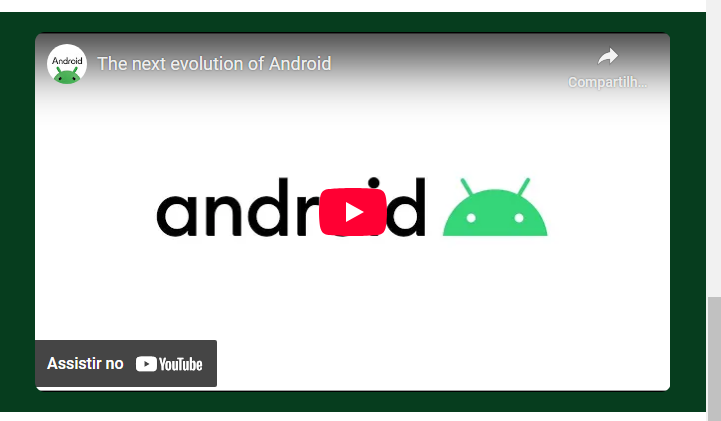
**CSS**

Passo 1 🡪 criar em estilo tudo que for div com a classe video.





Passo 2 🡪 colocar posicionamento absouto



***PROPRIEDADE DE ESTILO - POSITION:***

O posicionamento padrão de todo elemento no HTML é o "estático". Todo elemento estático é posicionado no canto superior esquerdo no corpo do documento ou do elemento onde ele está contido. Esta é a posição 0 de um elemento no corpo do documento.

A Propriedade de Estilo " position " é utilizada para determinar o posicionamento e hierarquia dos elementos na página do documento.

***PROPRIEDADES AUXILIARES DO POSITION:***

A Propriedade de Estilo "position" vem ACOMPANHADA e é usada em conjunto com quatro outras propriedades auxiliares (são propriedades e NÃO valores para position), que são:

top - *(topo)*

bottom - *(baixo)*

right - *(direita)*

left - *(esquerda)*

Essas quatro propriedades irão determinar a localização final do objeto, irão deslocar o elemento conforme for especificado.

Por exemplo, se usamos a propriedade "top" com o valor de 20px em um elemento, o mesmo vai se deslocar do topo num total de 20px para baixo.

***VALORES DA PROPRIEDADE POSITION:***

Para a propriedade position, é possível atribuir 5 valores, que são: static, relative, absolute, fixed e sticky.

Funções dos valores mais usados:

**static - (estático):**

O valor padrão da propriedade position é o static. Se não determinarmos nada, é esse valor que vai prevalecer por "debaixo dos panos". Ele não indica posicionamento algum, ou seja, é o valor para "nenhum posicionamento específico" do elemento. A função dele é manter a hierarquia de posicionamento conforme já estabelecido no documento HTML.

Isso significa que aquelas quatro outras propriedades já citadas (top, bottom, left e right) não funcionarão, não irão alterar a posição do elemento se usadas juntamente com esse valor... Visto que o position já está como "static" por padrão.

**relative - (relativo):**

O valor relative posiciona um elemento em relação à sua posição "normal"... Ou seja, as quatro propriedades auxiliares (top, bottom, left e right) vão fazer com que o elemento seja ajustado A PARTIR DA SUA POSIÇÃO ORIGINAL.

Após ter definido o tipo de position (relative nesse caso), a posição do elemento é definida através das quatro propriedades auxiliares do position: top, bottom, left e right. Que indicam a distância com relação ao topo, base, esquerda e direita, respectivamente, tomando como REFERÊNCIA DE POSICIONAMENTO (no caso do valor relative) a posição original do elemento. (se não configurarmos valores de posicionamento com as propriedades auxiliares, o elemento simplesmente fica no mesmo lugar)

**absolute - (absoluto):**

Um elemento com o valor absolute para position vai ser disposto, com as propriedades auxiliares, de forma relativa ao seu elemento pai... DESDE QUE ESSE ELEMENTO PAI TENHA UM POSITION COM UM VALOR DIFERENTE DE "STATIC" DEFINIDO!

Se o elemento pai não ter um position com um valor diferente de "static" definido, o valor absolute faz com que o elemento configurado " saia " da hierarquia estabelecida pelo HTML.

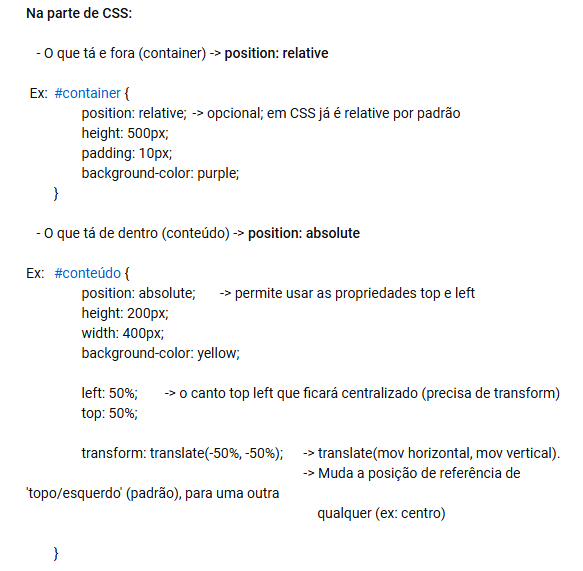
Um elemento com o position absolute é disposto em relação ao elemento de nível superior mais próximo dele que possui um position com um valor diferente de "static" definido... E como já foi dito, caso não exista nenhum elemento hierarquicamente de nível superior (pai ou "avô") que tenha um position sem ser o "static" definido, o position absolute será absoluto ao corpo da página (body).

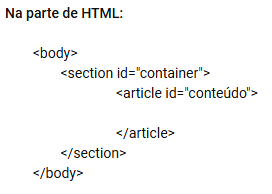
Na prática, em elementos do tipo filho com o pai sem a propriedade position, o valor absolute "transforma" filhos de elementos em FILHOS DA PÁGINA (body). Assim, é possível posicionarmos um elemento filho (transformado) em qualquer lugar da página (fora do pai), independentemente do que temos lá. Isso pode ser útil em casos que precisamos posicionar um alerta em nossa página, por exemplo, e não queremos "quebrar" o layout. ---

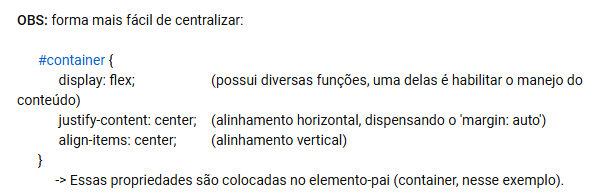
Dessa forma, note que o position absolute possui dois comportamentos diferentes, que se manifestam dependendo da configuração do elemento pai ou elemento de nível superior.

Nota: Perceba que o Guanabara configurou o filho da div (iframe) com a declaração "position: absolute" e o pai (a div) com a declaração "position: relative"... Dessa forma o elemento filho (iframe) é posicionado, com as propriedades auxiliares, em relação ao seu pai (div).

## ***CENTRALIZAÇÃO VERTICAL DE CAIXAS***







## ***COMANDOS DE TABELAS EM CSS 3***

### **position: relative**

Coloca-se a table na configuração posição relativa.

Para travar o titulo da tabela usa-se o código abaixo no cabeçalho da tabela.

     thead > tr > th {

        position: sticky;

        top: -1px;

        background-color: aqua;

     }

### **colspan = “ 2 ”**

<td colspan="2">B</td> 🡪 a célula B está ocupando as células de duas colunas.

### **rowspan = “ 3 ”**

<td rowspan="3">D</td> 🡪 a célula D está ocupando as células de três linhas.

# **JAVASCRIPT**

## ***Variáveis***

Em JS sempre que tiver um sinal de **=** você interpreta que recebe algo.

**Null** em JS significa que a variável está vazia.

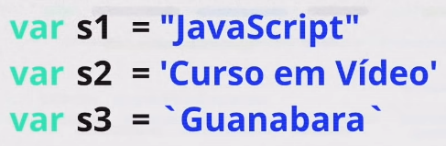
Variáveis ocupa a memória do computador.

**Var** ou **let** variável em JS.

Temos variáveis numéricas

Variáveis strings

Quando você coloca dados em uma variável, estamos atribuindo esse dado nesta variável.

Atenção, tem diferença entre aspas dupla “...”, apostrofe ‘...’ (aspas simples) e a crase `...`

O nome de cada variável é chamado de **identificadores**.

Regras

1. Podem começar com uma **letra, $** ou **\_**;
2. Não podem começar com números;
3. È possível usar letras ou números;
4. È possível usar acentos e símbolos;
5. Não podem conter espaços;
6. Não podem ser palavras reservadas pelo JS;
7. Maiúsculas e minúsculas fazem diferença;
8. Utilizar nomes coerentes para as variáveis. (evite tornar um programador alfabeto ou um programador contador);
9. Variáveis tipo number (são inteiro, números reais ponto flutuante float );
   1. Infinity
   2. NaN
10. Variaveis tipo strings (são cadeias de caracteres, pode ser letras e até numeros);
11. Variáveis boolean (true, false);
12. Tipos primitivos primordiais String, number e boolean, null, undefined, object, array, function.

## ***Manipulação de Dados***

## **Number(n) - Number.parseInt(n) – Number.parseFloat(n)**

Para converter strings para números:

* Number.parseInt(n) 🡪 Conversão para um número inteiro
* Number.parseFloat(n) 🡪 Conversão para um número real
* Number(n) 🡪 O JavaScript já define o tipo Int ou Float

## **String(n) – n.toString( )**

Para converter números para strings:

* n.String( ) 🡪 Conversão número para String
* String(n)

## ***COMANDOS EM JAVASCRIPT***

## ***//***

Comentários em uma única linha.

## ***/\* ... \*/***

Comentários em mais de uma linha (bloco).

## ***Window.alert ( ‘...’ );***

Abre uma caixa de dialogo para interação.

## ***Window.confirm ( ‘...’ );***

Abre uma caixa de diálogo para confirmação.

## ***Window.prompt ( ‘...’ );***

Abre uma caixa de diálogo para interação com o usuário. Os dados de números são strings, temos que converter.

## **typeof**

Mostra o tipo de variável é.



## **Concatenação**

Utiliza-se o sinal + e em nomes de variáveis não coloca os apostrofes.

Window.alert(‘É um grande prazer em te conhecer, ’ + nome + ‘!’)

## **Template strings**

Utiliza-se o sinal ${ } e o texto fica entre crase e em nomes de variáveis não coloca os apostrofes.

Var S = ‘JavaScript’

Window.alert(`Eu estou aprendendo ${S}`)

## **.lengh**

Var S = ‘JavaScript’

S.length 🡪 mostra o tamanho da string

## **.toUpperCase( )**

Var S = ‘JavaScript’

S.toUpperCase 🡪 transforma tudo em letras maiúsculas

## **.toLowerCase( )**

Var S = ‘JavaScript’

S.toLowerCase 🡪 transforma tudo em letras minúsculas

## **document.write(...)**

ex. document.write(`Seu nome tem ${nome.legth} letras`)

Este comando mantem na mesma linha.

## **document.write(...)**

ex. document.write(`Seu nome tem ${nome.legth} letras`)

Este comando mantem na mesma linha.

## ***Formatação de Números***

## **.toFixed(2)**

Para formatar números com 2 casas decimais. ex.

* var n1 = 1545.5
* n1.toFixed(2) 🡪 1545.50

## **.toFixed(2).replace( ‘.’ , ‘,’)**

Para formatar números com vírgula em vez de ponto. ex.

* var n1 = 1545.5
* n1.toFixed(2).replace( ‘ . ’ , ’ , ’ ) 🡪 1545,50

## **n1.toLocaleString(‘pt-BR’,{style: ‘currency’, currency:’BRL’})**

Formata o número para o formato do real R$

## **n1.toLocaleString(‘pt-BR’,{style: ‘currency’, currency:’USD’})**

Formata o número para o formato em dolar US$

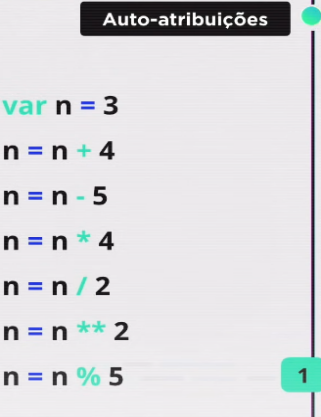
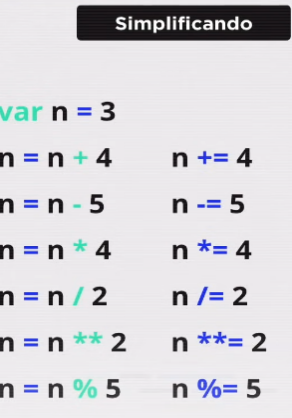
## **n1.toLocaleString(‘pt-BR’,{style: ‘currency’, currency:’EUR’})**

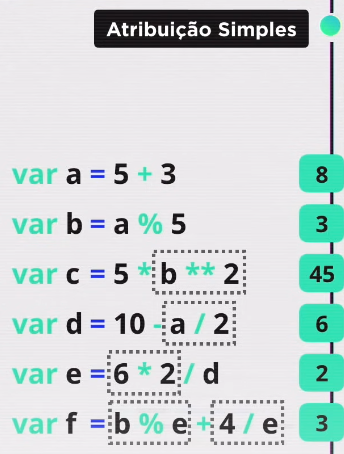
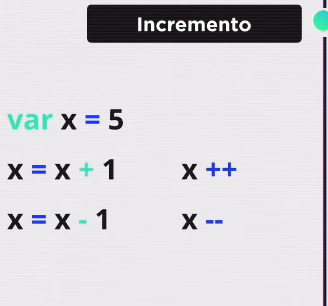
Formata o número para o formato em Euro

## ***Operadores***

Aritméticos, atribuição, relacionais, lógicos, ternário.

## **aritméticos ( + , - , \* , / , % , \*\* )**

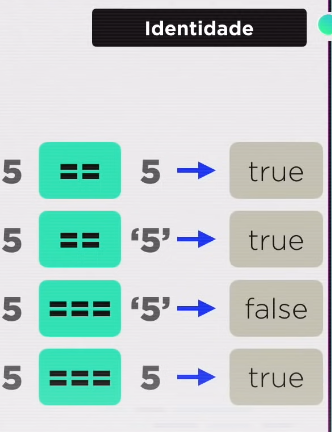


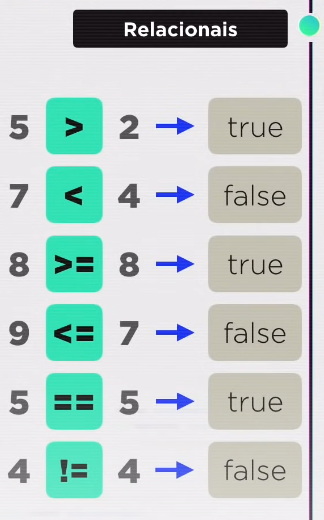


Tela de celular com texto preto sobre fundo branco

O conteúdo gerado por IA pode estar incorreto.

## **Relacionais ( > , < , >= , <= , == , != )**





## **Lógicos ( ! , && , || )**

Placa de computador com texto preto sobre fundo branco

O conteúdo gerado por IA pode estar incorreto.Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

O conteúdo gerado por IA pode estar incorreto.

Tela de celular

O conteúdo gerado por IA pode estar incorreto.Calculadora preta com letras brancas

O conteúdo gerado por IA pode estar incorreto.

## **Ternário ( ? : )**

Interface gráfica do usuário

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

## ***DOM***

Document Object Model.

Interface gráfica do usuário, Texto, Aplicativo

O conteúdo gerado por IA pode estar incorreto.Texto, Linha do tempo

O conteúdo gerado por IA pode estar incorreto.Diagrama

O conteúdo gerado por IA pode estar incorreto.

Por marca 🡪 **getElementsByTagName( )** [0... ] consegue selecionar mais de um elemento.

Por ID 🡪 **getElementByid( )** consegue selecionar um elemento com o ID especificado.

Por Nome 🡪 **getElementsByName( )[ ]** consegue selecionar mais de um elemento com o Nome especificado.

Por classe 🡪 **getElementsByClassName( )[ ]** consegue selecionar mais de um elemento com a classe especificada.

Por seletor 🡪 **querySelector( ) e querySelectorAll( )**

Var d = window.document.querySelector(‘div#msg’) 🡪 todo id é representada por uma #.

Var d = window.document.querySelector(‘div.msg’) 🡪 toda classe é representada por uma .

## **window.document.write(p1.innerText)**

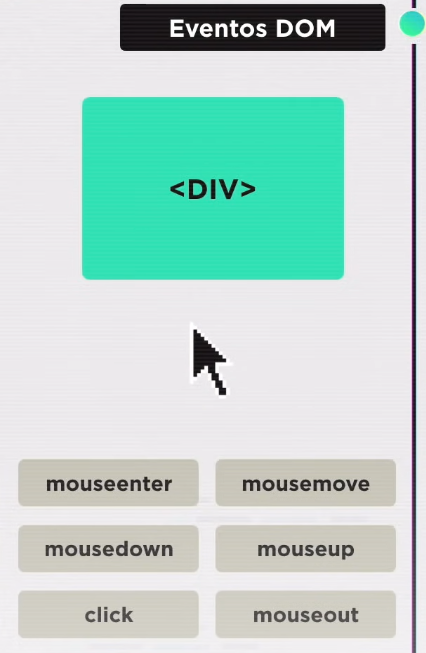
innerText 🡪 Escreve no document o texto sem a formatação que está no p1

## **window.document.write(p1.innerhtml)**

innerHTML 🡪 Escreve no document o texto com a formatação que está no p1

## ***Eventos DOM***

Eventos de mouse.

Mouseenter 🡪 Dispara um evento quando o mouse entra na div.

Mousemove 🡪 Dispara um evento quando o mouse move na div.

Mousedown 🡪 Dispara um evento quando o mouse é clicado e segurado na div.

Mouseup 🡪 Dispara um evento quando solta o click do mouse na div.

Click 🡪 Dispara um evento quando clica no mouse na div.

Mouseout 🡪 Dispara um evento quando o mouse sai da div.

## ***Funções JavaScript***

Conjunto de códigos que só vão ser executadas somente quando um evento ocorrer chamado de bloco.

Um bloco em Javascript é delimitado entre chaves { bloco }, esse bloco você nomeia ele com uma function ex.

function ação(param) {bloco}

## ***Condições***

Desvio condicional.

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

O conteúdo gerado por IA pode estar incorreto.Relógio digital com números

O conteúdo gerado por IA pode estar incorreto.

## **console.log (‘…’)**

Para rodar o JS no console, para mostrar no console usa-se o comando:

Interface gráfica do usuário, Texto

O conteúdo gerado por IA pode estar incorreto.

Interface gráfica do usuário, Texto, Aplicativo, Email

O conteúdo gerado por IA pode estar incorreto.

# ***Bibliografias***

### **Material curso Gustavo Guanabara**

#### 01 – História da Internet

<https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/01%20-%20Hist%C3%B3ria%20da%20Internet.pdf>

#### 02 – Como a Internet Funciona

<https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/02%20-%20Como%20funciona%20a%20Internet.pdf>

#### 03 – Como Funciona HTML e CSS

<https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/03%20-%20Como%20funciona%20HTML%20e%20CSS.pdf>

#### 04 – Primeiros Passos em HTML

<https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/04%20-%20Primeiros%20passos%20HTML.pdf>

#### 05 – Caracteres, parágrafos e quebras de linhas

<https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/05%20-%20Caracteres%2C%20par%C3%A1grafos%20e%20quebras%20de%20linha.pdf>

#### 06 – Imagens e Favicon

<https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/06%20-%20Imagens%20e%20Favicon.pdf>

#### 07 - Hierarquia de títulos

<https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/07%20-%20Hierarquia%20de%20t%C3%ADtulos.pdf>

#### 08 - Formatação de Textos

<https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/08%20-%20Formata%C3%A7%C3%A3o%20de%20Textos.pdf>

#### 09 – Listas HTML

<https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/09%20-%20Listas%20HTML.pdf>

#### 10 – Ligações em toda parte

<https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/10%20-%20Liga%C3%A7%C3%B5es%20em%20toda%20parte.pdf>

#### 11 – [Imagens dinâmicas, áudios e vídeos](https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/11%20-%20Imagens%20din%C3%A2micas%2C%20%C3%A1udios%20e%20v%C3%ADdeos.pdf)

<https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/11%20-%20Imagens%20din%C3%A2micas%2C%20%C3%A1udios%20e%20v%C3%ADdeos.pdf>

#### 12 – [Trabalhando com estilos](https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/12%20-%20Trabalhando%20com%20estilos.pdf)

<https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/12%20-%20Trabalhando%20com%20estilos.pdf>

#### 13 – [Cores](https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/12%20-%20Trabalhando%20com%20estilos.pdf)

<https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/13%20-%20Cores.pdf>

#### 14 – [Fontes](https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/12%20-%20Trabalhando%20com%20estilos.pdf)

<https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/14%20-%20Fontes.pdf>

#### 15 – [Seletores personalizados](https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/12%20-%20Trabalhando%20com%20estilos.pdf)

<https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/15%20-Seletores%20personalizados.pdf>

#### 16 – [Modelos de Caixas](https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/12%20-%20Trabalhando%20com%20estilos.pdf)

<https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/16%20-%20Modelo%20de%20Caixas.pdf>

#### 17 – [Primeiro mini-projetos](https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/12%20-%20Trabalhando%20com%20estilos.pdf)

<https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/17%20-%20Primeiro%20mini-projeto.pdf>

#### 18 – [Aprendendo Git e GitHub](https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/12%20-%20Trabalhando%20com%20estilos.pdf)

<https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/18%20-%20Aprendendo%20Git%20e%20GitHub.pdf>

#### 19 – [Imagens de Fundo](https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/12%20-%20Trabalhando%20com%20estilos.pdf)

<https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/19%20-%20Imagens%20de%20Fundo.pdf>

#### 20 – [Mini-projeto Cordel](https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/12%20-%20Trabalhando%20com%20estilos.pdf)

<https://github.com/gustavoguanabara/html-css/blob/master/aulas-pdf/20%20-%20Mini-projeto%20Cordel.pdf>

### [**Referêcias**](https://developer.mozilla.org/pt-BR/) **on-line**

#### 01 - <https://developer.mozilla.org/pt-BR/> (MDN – Mozilla developer network)

#### 02 - <https://www.w3.org/> (W3C – Referência oficial inglês)

#### 03 - <https://html.spec.whatwg.org/multipage/> (WHATWG Living Standard)

#### 04 - <https://www.w3schools.com/> (W3Schools – muito bom)

#### 05 - <https://emojipedia.org/> (site para baixar emoji)

### **Livros**

#### 01 – HTML5 Mark Pilgrim (O´Reilly) - Portugues

#### 02 – CSS Eric Meyer (O´Reilly) - Portugues

#### 03 – Flexbox in CSS Estelle Weyl (O´Reilly)

#### 04 – Grid Layout in CSS Eric Meyer (O´Reilly)

#### 05 – HTML5 e CSS3 Elizabeth Castro e Bruce Hyslop (Alta Books) - Portugues

#### 06 – HTML e CSS Jon Duckett (Alta Books) – Portugues - bom para comprar

#### 07 – HTML e CSS Elisabeth Robson e Eric Freeman (Alta Books) – Portugues – Iniciante muito bom

#### 08 – Crie seu próprio site Nate Cooper e Kim Gee (Novatec) – Portugues – tipo historia em quadrinhos

#### 09 – HTML5 Maurício Samy Silva (Novatec)

#### 10 – CSS3 Maurício Samy Silva (Novatec)

#### 11 – HTML5 e CSS3 Maurício Samy Silva (Novatec)

#### 12 – CSS Grid Layout Maurício Samy Silva (Novatec)

### **Programas**

#### 01 - Chrome Web Store (<https://chromewebstore.google.com/>) ( loja de extensões)

#### 02 - web developer (serve para desativar CSS, JavaScript etc)

#### 03 – Visual Studio Code (<https://code.visualstudio.com/>)

***Configurações***

*Português, procurar em extensões português (escolher da Microsoft)*

*Arquivo 🡪 Preferencias 🡪 Configurações 🡪 procurar por Word Wrap 🡪 on (quebra a linha do*

*código )*

*Arquivo 🡪 Preferencias 🡪 Configurações 🡪 fontSize 🡪 14 (tamanho da fonte)*

*Arquivo 🡪 Preferencias 🡪 Configurações 🡪 Workbench: Color Theme 🡪 Default Dark+*

*Clicando entre as aspas a tecla control+espaço (já aparece a lista dos arquivos).*

*Em um texto selecionado, usar a tecla control+shift+p (selecionar Emmet: Envelope com abreviatura) apertar enter e digitar a tag e apertar enter ele já coloca a tag entre o texto selecionado.*

Quem não quer ficar colocando o pt-br toda hora na estrutura padrão do html criado pelo snippet pode configurar uma variável de ambiente indo em Arquivo > Preferências > Configurações, depois pesquisa por "variáveis emmet" e depois clica em adicionar item e cria a variável lang e o valor coloca pt-br. Aí todas as vezes que pressionar o ! + Enter ele já vai criar o código padrão com o parâmetro lang="pt-br"

#### 04 – Gimp

<https://www.gimp.org/> (programa para edição de imagem tipo photoshop mas freeware)

#### 05 – HandBrake

<https://handbrake.fr/> (converter os vídeos - programa freeware)

#### 06 – Mockflow

<https://mockflow.com/> (Fazer estrutura de site muito bom)

#### 07 – Windiw Resizer

Extensão do google crome (mostra o tamanho da tela)

Em configurações, permitir acesso aos URLs do arquivo

#### 08 – Git

Fazer o download

**ATENÇÃO** :Primeiro tem que instalar o Git antes do GitHub e ter o Visual Studio Code instalado.

Selecionar o editor para usar (Visual Studio Code)

Reiniciar a maquina.

#### 09 – GitHub desktop

Fazer o download

Instalar o GitHub Desktop

Entrar no GitHub.com (tem que ter já feito a conta)

#### 10 – Node DS

<https://nodejs.org/pt> (Executar o Java Script em qualquer lugar)

Para limpar a tela usar control+L

#### 11 – Live Server

Extensão do Visual Studio Code ( )

#### 12 – Node.js Exec

Extensão do Visual Studio Code ( ) 🡪 executa o Javascript pela tecla F8 no visual stud code.

### **Imagens Direitos Autorais**

#### 01 – Google imagens – Edge, Safari, chrome etc

1. Pesquisar a imagem
2. Entrar em ferramentas e selecionar direitos de uso

#### 02 – Wikimedia

<https://commons.wikimedia.org/wiki/Main_Page>

#### 03 – Unsplash

<https://unsplash.com/pt-br/> (fazer a busca em inglês)

#### 04 – Pexels

<https://www.pexels.com/pt-br/> (fazer a busca em inglês)

### **Ícone de favoritos**

#### – Iconarchive

<https://www.iconarchive.com/> (Baixar arquivo ICO)

#### – Favicon.cc

<https://www.favicon.cc/> (Criar ícones ICO)

#### – Favicon.io

<https://favicon.io/> (Criar ícones a partir de Texto 🡪 ICO, PNG 🡪 ICO e um favicon 🡪 ICO)

### **Efeitos e músicas**

#### – YouTube

Acessar o painel de controle 🡪 YouTube Studio, e clicar em biblioteca de áudio.

Pesquisar ou filtrar biblioteca e escolher entre:

***Atribuição necessária*** 🡪 você deve colocar a atribuição de licença no seu código

***Atribuição não é necessária*** 🡪 totalmente livre para uso sem restrições

### **Vídeos**

#### 01 – Pexels

<https://www.pexels.com/pt-br/> (fazer a busca em inglês)

### **Listas de media types**

Se você quer saber o que escrever dentro do atributo type de uma âncora de hypertext, consulte a lista oficial da IANA.org disponível no link abaixo.

<https://www.iana.org/assignments/media-types/media-types.xhtml>

### **Cores**

#### 01 – Adobe Color

<https://color.adobe.com/pt/>

#### 02 – Paletton

<https://paletton.com/#uid=1000u0kllllaFw0g0qFqFg0w0aF>

#### 03 – Coolors

<https://coolors.co/>

#### 04 – ColorZilla

(<https://chromewebstore.google.com/>) ( loja de extensões google chrome)

Serve para mostrar e copiar o código da cor de qualquer site

### **Fontes**

#### 01 – Google Fonts

<https://fonts.google.com/>

#### 02 – dafont.com

<https://www.dafont.com/pt/>

#### 03 – Fonts Ninja

(<https://chromewebstore.google.com/>) ( loja de extensões google chrome)

Serve para descobrir as fontes de textos usada em sites.

#### 04 – What Font is

<https://www.whatfontis.com/>

Serve para descobrir as fontes de textos usada em imagens em sites.

#### 05 – Font Squirrel

<https://www.fontsquirrel.com/>

Serve para descobrir as fontes de textos usada em imagens em sites.

#### 06 – My Fonts

<https://www.myfonts.com/pages/tags/my-fonts>

Serve para descobrir as fontes de textos usada em imagens em sites.